# 使用智慧卡之通行碼身分鑑別協定

廖鴻圖
世新大學資訊管理學系

林建福
德明技術學院資訊管理學系

蕭麗齡
世新大學資訊管理學系

鄭惠菱
世新大學資訊管理學系

## 摘要

隨著網際網路（Internet）的快速興起，使用者已逐漸透過網路來完成各種交易，因此如何在分散式網路上，使遠端伺服器確認使用者的身分與使用權限，則變成相當重要的議題。本文提出一個兼具安全性與完整性的以通行碼為基礎身分鑑別機制，並滿足人性化需求、雙向鑑別、低計算與通訊成本等多項特性。此外，本機制透過鑑別的過程，同時也產生一把會議金鑰，以確保鑑別後通訊訊息的私密性與完整性，並利用公開金鑰密碼技術來解決使用者註冊階段時安全通道不合理假設的問題。

**關鍵字：**身分鑑別、通行碼、單向雜湊函數、密碼學。

# A Password-based Authentication Scheme Using Smart Cards

Horng-Twu Liaw
Department of Information Management, Shih Hsin University

Jiann-Fu Lin
Department of Management Information System, Takming College

Li-Lin Hsiao
Department of Information Management, Shih Hsin University

Hui-Ling Cheng
Department of Information Management, Shih Hsin University

## Abstract

Owing to the rapid development of the Internet, each user can finish various kinds of trade via the network. Therefore, it is an important issue that how to confirm the identity of user and user's access rights for any remote server. This paper proposes a secure password-based authentication scheme that satisfies several secure properties including user friendly, mutual authentication, lower computation and communication cost. Besides, this scheme produces the session key through the processes of authentication to ensure privacy and integrity, and solves the unreasonable assumption problem of the secure channel.

Key words: Authentication, Password, One-way hash function, Cryptography.

# 1. Introduction

In a traditional authentication scheme, each user becomes the legal user after register the server (*S* for short) and has a corresponding couple of the user's identity (*ID* for short) and secret password (*PW* for short). The *S* stores the same information in the verification table. Hence, if a legal user wants to obtain resources or services from the *S*, he/she must send his/her own *ID* and *PW* to the *S* through the insecure network. At first, the *S* would confirm the *ID* whether it exited in the verification table or not, this part is called identification. The *S* would continue to confirm the *PW* stored in the verification table which was corresponding to *ID* whether it and the *PW* that user send are the same or not, this part is called authentication（Chang, Lin, Chiang, 2000）. The Lamport (1981) proposed a hash-function based authentication scheme, which uses a verification table to store each user's *ID* and the corresponding one-way hashing value of *PW* in the *S* to enhance security. The Shimizu (1990) proposed a conception of one-time-password that has divided the password into the user-password and verify-password. Each the user-password is for the memorizing *PW*, and the verify-password is for the *PW* stored in the verification table. Each user could utilize the user-password to generate the verify-password for the purpose of login the *S* (Ku, Tsai, Chen, 2002). To prevent off-line password guessing attack, the Sandirigama (2000) proposed a simple and secure password authentication protocol. Recently, the Wang (2004) proposed a scheme that utilizes timestamp to prevent replay attack. But, timestamp would incur the serious time-synchronization problem that the user's time and the *S*'s time must be the same or must differ in a reasonably small range. Therefore, this paper proposed a secure password-based authentication scheme that adopts Nonce to solve this problem and satisfy several secure properties including user friendly, mutual authentication, lower computation and communication cost. And, this proposed scheme uses smart card to store some authentication information. Embedded into the smart card are a microprocessor and other circuitry that can be programmed to do a wide variety of tricks. Besides, this proposed scheme produces the session key through the processes of authentication to ensure privacy and integrity, and solve the unreasonable assumption problem of the secure channel with Public-Key Cryptography.

In Section 2 and 3, the relative schemes proposed by Chen (2003) and Wang (2004) would be introduced, respectively. Then an improved authentication scheme would be proposed to satisfy several secure properties and then the security and integrity would be analysis in the Section 5. Finally, there would be a brief conclusion.

# 2. The Chen's Scheme

In 2000, the Sandirigama (2000) proposed a SAS protocol (a simple and secure password authentication protocol) and claimed it would prevent man-in-the-middle attack. However, a SAS protocol suffers from DoS attack, the replay attack and stolen-verifier attack (Lin, Sun, Hwang, 2001; Ku, Chen, 2002; Ku, Chen, Tsai, 2003). Therefore, the Lin et al. (2001) proposed an OSPA protocol (an optimal strong-password authentication protocol) to solve these problems. The Tsuji et al. (2003) showed that OSPA protocol is vulnerable to impersonation attack. Simultaneously, the Lin (2003) showed that OSPA protocol is also vulnerable to stolen-verifier attack. The Ku et al. (2003) showed that OSPA protocol is vulnerable to man-in-the-middle attack and stolen-verifier attack. Then, the Chen (2003) proposed an improvement scheme to arrive at security. Independently, the Wang (2004) proposed an improved scheme that is based on the SAS and OSPA protocols to satisfy security and integrity.

Before the schemes were described by this paper, the notations would be defined first that $x$ is the secret value was maintained by the $S$, $h()$ is a kind of one-way hash function, $\oplus$ is the exclusive-OR operation, $=>$ stands for the secure channel, $\rightarrow$ stands for the insecure channel, the $||$ is an operation that connects two strings, $N_c$ is represent a random number, the $F()$ is a kind of one-way hash function used to generate the session key, $T_1$ and $T_3$ are timestamps stand for the time for the smart card's and the server's respectively. The $T_2$ and $T_4$ stand for the received time for the smart card and the server's respectively.

In this section, the Chen's scheme would be described. There are two phases in the Chen's scheme: the registration phase and the authentication phase. The procedures of the registration phase are illustrated in Figure 1, and the procedures of the authentication phase are showed in Figure 2.

## 2.1 The Register phase

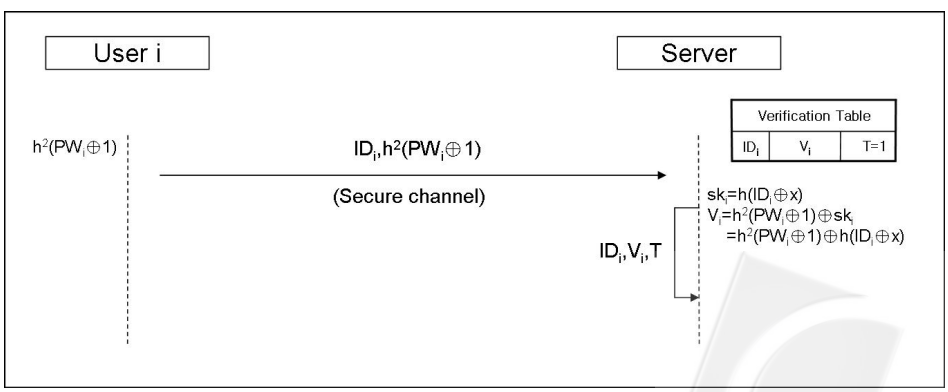


Figure 1: the Chen's scheme – the registration phase

1. *$U_i$ (User i) =>S (Server): $ID_i$, $h^2(PW_i \oplus 1)$*

   (1) The $U_i$ freely chooses $ID_i$ and $PW_i$ to register his/her identity to the $S$.

   (2) Then the $U_i$'s system would compute the $h^2(PW_i \oplus 1)$ with the $U_i$'s $PW_i$ and the initial value $T=1$.

   (3) After that, the $U_i$'s would send the registration information $ID_i$ and $h^2(PW_i \oplus 1)$ to the $S$ through the secure channel.

2. *S→verification table: $ID_i$, $V_i$, h()*

   (1) After the $S$ received the registration information from the $U_i$, the $S$ would compute the equations following:

   $sk_i = h(ID_i \oplus x)$

   $V_i = h^2(PW_i \oplus 1) \oplus sk_i$

   (2) Finally, the $S$ would insert the verifier information $ID_i$, $V_i$ and $T=1$ into the verification table.

## 2.2 The Authentication phase (nth login process)

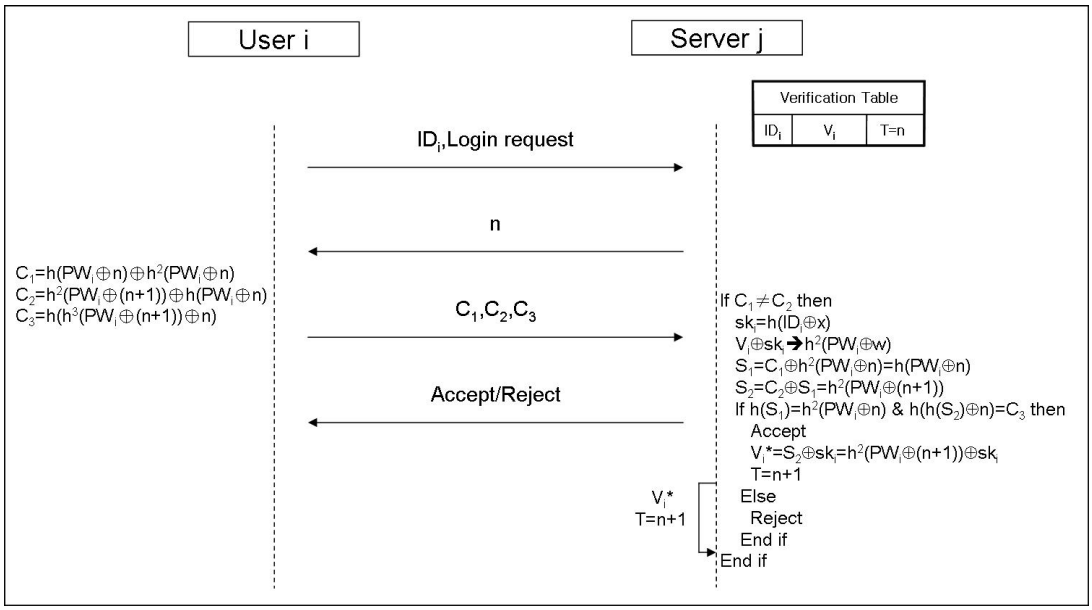


Figure 2: the Chen's scheme – the authentication phase

1. *$U_i$→S: $ID_i$, Login Request*

   (1) When the $U_i$ wants to login the $S$ and obtain the services or resources from the $S$, he has to send the login information $ID_i$ and login request to the $S$.

2. $S \rightarrow U_i$: $n$

(1) After the $S$ received the login information from the $U_i$, the $S$ would response to the $U_i$ with $U_i$'s nth sequential number $T=n$.

3. $U_i \rightarrow S$: $C_1$, $C_2$, $C_3$

(1) After that, the $U_i$ would compute the equations following:
$C_1 = h(PW_i \oplus n) \oplus h^2(PW_i \oplus n)$
$C_2 = h^2(PW_i \oplus (n+1)) \oplus h(PW_i \oplus n)$
$C_3 = h(h^3(PW_i \oplus (n+1)) \oplus n)$

(2) The $U_i$ sends the authentication information $C_1$, $C_2$ and $C_3$ to the $S$ through the insecure channel.

(3) The $S$ makes sure $C_1 \neq C_2$. If they do not hold, the $S$ would reject the login request from the $U_i$. Otherwise, the $S$ would use the verifier $V_i$ and $T=n$ that were stored in the verification table and the $C_1$, $C_2$ and $C_3$ to compute the equation following:
$sk_i = h(ID_i \oplus x)$
$V_i \oplus sk_i \rightarrow h^2(PW_i \oplus n)$
$S_1 = C_1 \oplus h^2(PW_i \oplus n) = h(PW_i \oplus n)$
$S_2 = C_2 \oplus S_1 = h^2(PW_i \oplus (n+1))$

(4) If the $h(S_1)$ does not equal to the $h^2(PW_i \oplus n)$ or $h(h(S_2) \oplus n)$ does not equal to $C_3$, the $S$ would reject the login request from the $U_i$. Otherwise, the $S$ would accept the login request and compute the equation following:
$V_i^* = S_2 \oplus sk_i$
$= h^2(PW_i \oplus (n+1)) \oplus sk_i$

(5) Then the $S$ would replace $V_i$ with $V_i^*$, and set $T=n+1$ for the $U_i$ next authentication.

# 3. The Wang's Scheme

In this section, the Wang's scheme would be described. There are two phases in the Wang's scheme: the registration phase and the authentication phase. The procedures of the registration phase are described in Figure 3, and the procedures of the authentication phase are illustrated in Figure 4.

## 3.1 The Registration phase

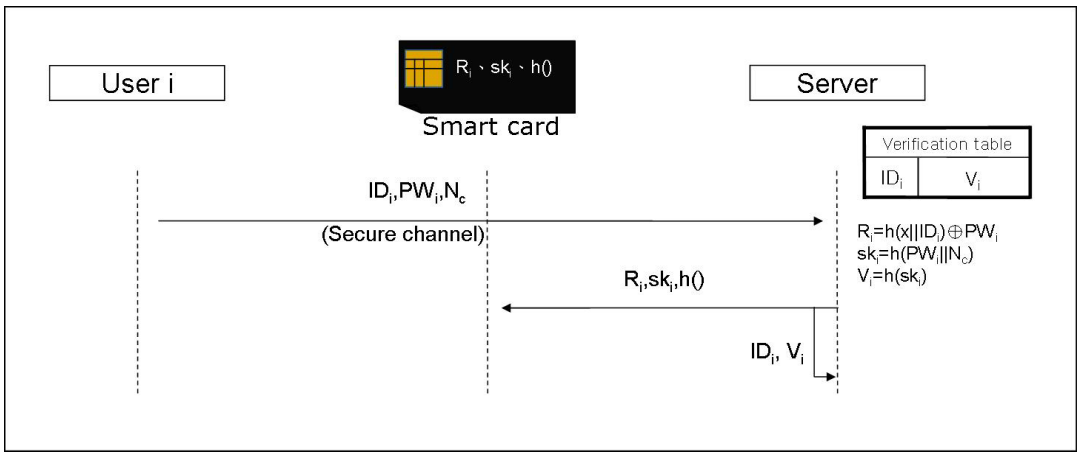


Figure 3: the Wang's scheme – the registration phase

1. *$U_i$—>S: $ID_i$, $PW_i$, $N_c$*

   (1) The $U_i$ would send his $ID_i$, $PW_i$ and a random number $N_c$ that is generated by the $U_i$'s system to register his identity to the $S$ through the secure channel.

2. *S→$U_i$: $SC_i$ (Smart Card)*

   (1) After the $S$ has received the registration information from the $U_i$, the $S$ would compute the equations following:

   $R_i=h(x||ID_i) \oplus PW_i$

   $sk_i=h(PW_i||N_c)$

   $V_i=h(sk_i)$

   (2) Then, the $S$ would write the $R_i$, $sk_i$ and $h()$ into the $SC_i$ and insert the verifier information $ID_i$ and $V_i$ into the verification table.

   (3) At last, the $S$ would send the $SC_i$ to the $U_i$.

## 3.2 The Login phase and Authentication phase

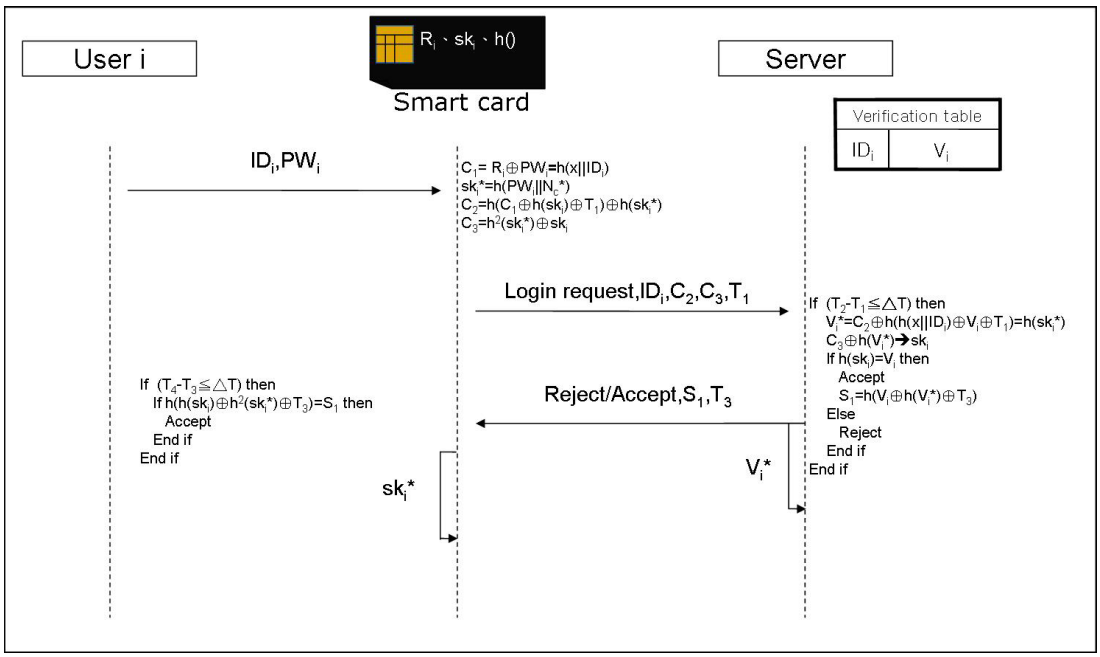


Figure 4: the Wang's scheme – the login and authentication phase

1. $SC_i \rightarrow S$: $ID_i$, $C_2$, $C_3$, $T_1$

   (1) When the $U_i$ wants to login the $S$ and obtain the services or resources, he has to insert his smart card to the device and input his $ID_i$ and $PW_i$, and then the $SC_i$ would compute the equations following:

   $$C_1 = R_i \oplus PW_i$$
   $$= h(x||ID_i)$$

   (2) After that, the $SC_i$ would generate a random number $N_c$* and use the login time as $T_1$ to compute the equations following:

   $$sk_i^* = h(PW_i||N_c^*)$$
   $$C_2 = h(C_1 \oplus h(sk_i) \oplus T_1) \oplus h(sk_i^*)$$
   $$C_3 = h^2(sk_i^*) \oplus sk_i$$

   (3) Finally, the $SC_i$ would send the information $ID_i$, $C_2$, $C_3$ and $T_1$ to the $S$.

2. $S \rightarrow SC_i$: $S_1$, $T_3$

   (1) After the $S$ has received the login request and information $ID_i$, $C_2$, $C_3$ and $T_1$, the $S$ would use the received time as $T_2$ and check $T_2 - T_1 \leqq \triangle T$ ($\triangle T$ stands for the valid time interval for transmission delay). Then, the $S$ would use the $U_i$'s $ID_i$,

secret number $x$, $T_1$ and $V_i$ that would be stored in the verification table to compute the equations following:

$V_i^*=C_2 \oplus h(h(x||ID_i) \oplus V_i \oplus T_1)$

$=h(sk_i^*)$

$C_3 \oplus h(V_i^*) \rightarrow sk_i$

(2) If the $h(sk_i)$ does not equal to the $V_i$ that be stored in the verification table, the $S$ would reject the login request from the $U_i$. Otherwise, the $S$ would accept and compute the equations following:

$S_1=h(V_i \oplus h(V_i^*) \oplus T_3)$

(3) After that, the $S$ would return the $S_1$ and $T_3$ to the $U_i$, and replace $V_i$ with $V_i^*$ for the $U_i$ next authenticate.

3. $SC_i \rightarrow SC_i$: $sk_i^*$

(1) The $U_i$ would use the received time as $T_4$ and check $T_4-T_3 \leqq \triangle T$ ($\triangle T$ stands for the valid time interval for transmission delay). If it does not hold, the $U_i$ would reject the services or resources from the $S$. Otherwise, the $U_i$ would use the $sk_i$, $N_c^*$ and $T_3$ to compute the equations following:

$h(h(sk_i) \oplus h^2(sk_i^*) \oplus T_3)$

(2) If the $h(h(sk_i) \oplus h^2(sk_i^*) \oplus T_3)$ does not equal to the received value $S_1$, the $U_i$ would reject the services or resources from the $S$. Otherwise, the $U_i$ would achieve the mutual authentication with the $S$ and restore the $sk_i^*$ into the $SC_i$, and the information would be used as the next time authentication information.

# 4. Our Improved Scheme

In this section, the proposed scheme would be described. There are four phases in the proposed scheme: the registration phase, the login phase, the authentication and the key agreement phase, and the password change phase. For contrasting with the previous scheme, the nations in the proposed scheme would be defined as previous scheme. Simultaneously, for solved unreasonable assumption of the secure channel, the proposed scheme would adopt the Public Key Cryptosystem to protect the register information. For this reason, the proposed scheme shall use the public key and the private key of the server. However, the issues which about how to generate keys and how to confirm validly of the key does not discuss in our research. Besides, in order to securely store secret number x in the server, the proposed scheme will adopt a kind of hardware equipment as storage device. When the server retrieved the secret number $x$ from the hardware device, it must be through the encrypting technology that is provided by hardware device (Network Security Technology Company Limited 2005). Because the hardware device does it, this shall not affect the proposed scheme's efficiency.

## 4.1 The Registration phase

The registration phase is invoked only once, when a new user requests to register with server. The procedures of this phase are described in Figure 5.

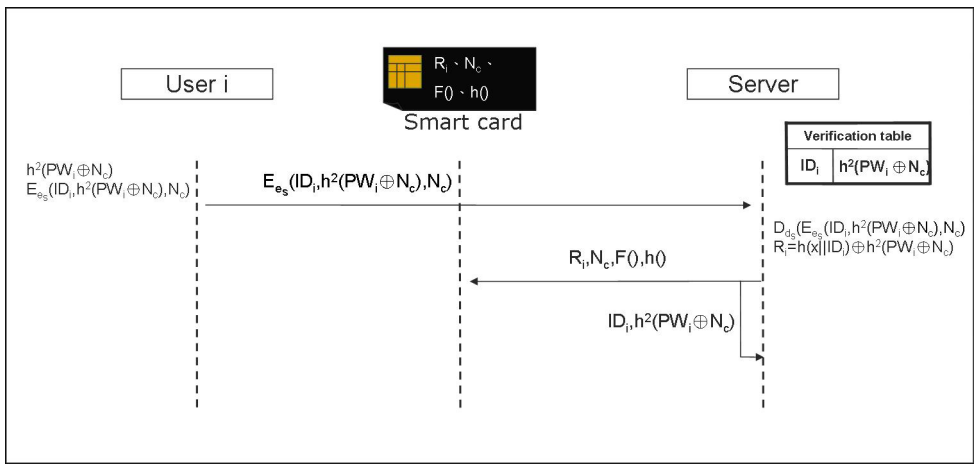


Figure 5: Our scheme – the registration phase

1. $U_i \rightarrow S$: $E_{es}(ID_i, h^2(PW_i \oplus N_c), N_c)$

   (1) The $U_i$ freely chooses $ID_i$ and $PW_i$ to register his identity to the $S$.

   (2) Then the $U_i$'s system would generate a random number $N_c$ and compute the $h^2(PW_i \oplus N_c)$.

   (3) After that, the $U_i$ would use the $S$'s public key $e_s$ to encrypt the register information $ID_i$, $h^2(PW_i \oplus N_c)$ *and* $N_c$. Finally, the $U_i$ sends the ciphertext to the $S$ through the insecure channel.

2. $S \rightarrow SC_i$: $R_i$, $N_c$, $F()$, $h()$

   (1) After the $S$ has received the register information, the $S$ would decrypt it to obtain the $ID_i$, $h^2(PW_i \oplus N_c)$ and $N_c$. And, the $S$ would use these to compute the $R_i$:
   $R_i = h(x||ID_i) \oplus h^2(PW_i \oplus N_c)$

   (2) Then, the $S$ would write the $R_i$, $N_c$, $F()$ and $h()$ into the $SC_i$ and insert the verifier information $ID_i$ and $h^2(PW_i \oplus N_c)$ into the verification table.

   (3) At last, the $S$ would send the $SC_i$ to the $U_i$.

## 4.2 The Login phase

The login, authentication, and key agreement phases are invoked whenever $U_i$ wants to login to a server to access the resources, and the server authenticates the user whether it is legal or not. The procedures of this phase are illustrated in Figure 6.

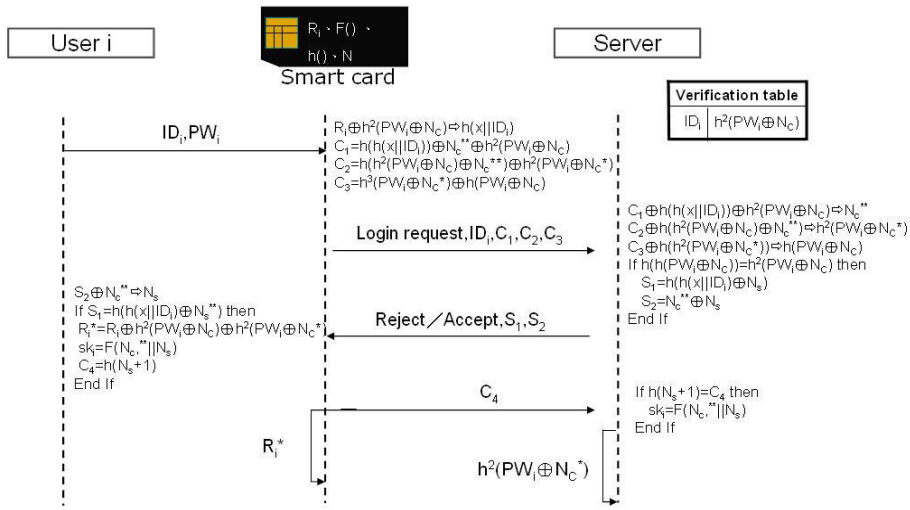


Figure 6: Our scheme – the login, authentication, and key agreement phase

1. *$SC_i \rightarrow S$: $ID_i$, $C_1$, $C_2$, $C_3$*

   (1) When the $U_i$ wants to login the $S$ for obtain the services or resources, he/she has to insert his/her $SC_i$ to the device and input his $ID_i$ and $PW_i$.

   (2) Then, the $SC_i$ would use the $U_i$'s $PW_i$ and the information ($N_c$ and $R_i$) which stored in the $SC_i$ to compute the equation following:

   $R_i \oplus h^2(PW_i \oplus N_c) \rightarrow h(x||ID_i)$.

   (3) After that, the $SC_i$ would generate the random number $N_c$* and $N_c$** to compute the equations following:

   $C_1 = h(h(x||ID_i)) \oplus N_c^{**} \oplus h^2(PW_i \oplus N_c)$

   $C_2 = h(h^2(PW_i \oplus N_c) \oplus N_c^{**}) \oplus h^2(PW_i \oplus N_c^{*})$

   $C_3 = h^3(PW_i \oplus N_c^{*}) \oplus h(PW_i \oplus N_c)$

   (4) Finally, the $SC_i$ would send the information $ID_i$, $C_1$, $C_2$ and $C_3$ to the $S$.

## 4.3 The Authentication and the Key Agreement phase

1. *$S \rightarrow SC_i$: $S_1$, $S_2$*

   (1) After the $S$ has received the login request and the information $ID_i$, $C_1$, $C_2$ and $C_3$, the $S$ would use the $U_i$'s $ID_i$, secret number $x$ and $h^2(PW_i \oplus N_c)$ that would be stored in the verification table to compute the equations following:

   $C_1 \oplus h^2(x||ID_i) \oplus h^2(PW_i \oplus N_c) \rightarrow N_c^{**}$

   $C_2 \oplus h(h^2(PW_i \oplus N_c) \oplus N_c^{**}) \rightarrow h^2(PW_i \oplus N_c^{*})$

   $C_3 \oplus h(h^2(PW_i \oplus N_c^{*})) \rightarrow h(PW_i \oplus N_c)$

(2) If the $h(h(PW_i \oplus N_c))$ does not equal to the $h^2(PW_i \oplus N_c)$ that would be stored in the verification table, the $S$ would reject the login request from the $U_i$. Otherwise, the $S$ would return the $S_1=h(h(x||ID_i) \oplus N_s)$ and $S_2=N_c{**} \oplus N_s$.

2. $U_i \rightarrow S: C_4$

(1) The $SC_i$ received the information $S_1$ and $S_2$ that returned by the $S$, and then the $SC_i$ to compute the equations following:

$S_2 \oplus N_c{**} \rightarrow N_s$

$h(h(x||ID_i) \oplus N_s))$

(2) If the $h(h(x||ID_i) \oplus N_s))$ does not equal to the received information $S_1$, the $U_i$ would reject the services and resources that the $S$ provides. It would ensure the identity of the $S$ is legal to arrive at the mutual authentication and avoid DoS attack.

(3) If the $h(h(x||ID_i) \oplus N_s))$ equals to $S_1$, then the $SC_i$ would compute the next time authentication information $R_i{*}$. Simultaneously, the $SC_i$ would use the $N_c{**}$, $N_s$ and $F()$ that would be stored in the $SC_i$ to compute the session key $sk_i$. And, the $SC_i$ returns $C_4$ to the $S$.

$R_i{*}=R_i \oplus h^2(PW_i \oplus N_c) \oplus h^2(PW_i \oplus N_c{*})$

$sk_i=F(N_c{**}||N_s)$

$C_4=h(N_s+1)$

(4) If the $h(N_s+1)$ is equals to $C_4$ that the $S$ has received, then the $S$ would accept and replace the next time authentication information $h^2(PW_i \oplus N_c{*})$ into the verification table. At last, the $S$ would use the $N_c{**}$, $N_s$ and $F()$ to compute the session key $sk_i=F(N_c{**}||N_s)$ as the same as the $U_i$ computed.

(5) For preventing the man-in-the-middle attack, the $SC_i$ and the S would restore the next time authentication information $R_i{*}=h^2(PW_i \oplus N_c{*})$ into the $SC_i$ and the verification table when the $U_i$ logouts from S, respectively.

## 4.4 The Change Password phase

This phase is invoked whenever the $U_i$ requests to change the original password $PW_i$ to new password $PW_i^*$. The procedures of this phase are illustrated in Figure 7.

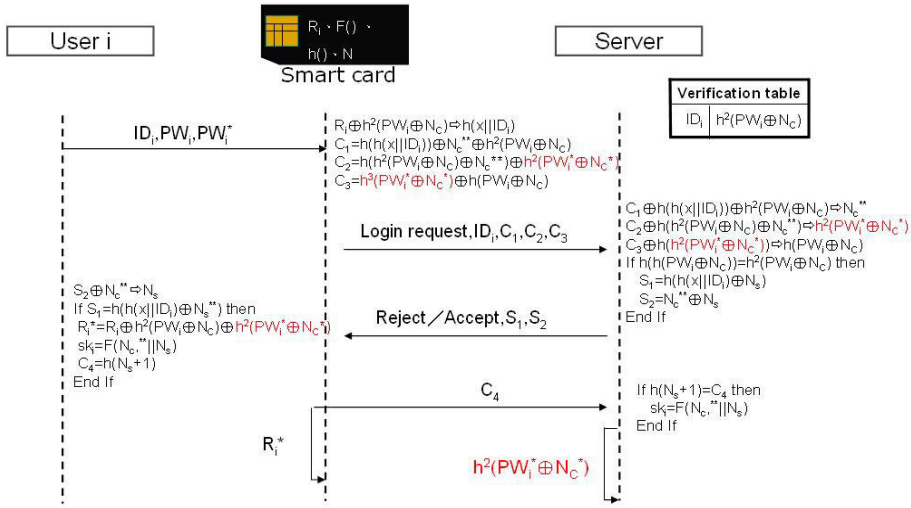


Figure 7: Our scheme – the change password phase

The proposed scheme preserves the next authentication information with authentication table that is maintained by the *S*.

Therefore, the change password phase needs to adopt on line to arrive. When the $U_i$ wants to change his original password $PW_i$ to new password $PW_i^*$, he/she has to insert his/her $SC_i$ into the device and input his $ID_i$, $PW_i$, and $PW_i^*$. Its way of doing things is the same as the login phase and the authentication and the key agreement phase. Hence, we no longer explain it once.

# 5. The Security and Integrality Analysis

## 5.1 The security analysis

1. Prevent DoS attack: while the $U_i$ sends the authentication information to the *S* to login, the value $C_2$ and $C_3$ include the next time authentication information. Hence, the *S* would use the $U_i$'s $ID_i$, secret number $x$ and $h^2(PW_i \oplus N_c)$ that would be stored in the verification table to obtain $N_c$** through compute the equation $C_1 \oplus h^2(x||ID_i) \oplus h^2(PW_i \oplus N_c)$. Similarly, the *S* could use the value $N_c$**, $C_2$ and $h^2(PW_i \oplus N_c)$ that would be stored in the verification table to obtain the next time authentication information $h^2(PW_i \oplus N_c^*)$ through compute the equation $C_2 \oplus h(h^2(PW_i \oplus N_c) \oplus N_c^{**})$. Then, the *S* could obtain this time authentication information $h(PW_i \oplus N_c)$ through compute the equation $C_3 \oplus h(h^2(PW_i \oplus N_c^*))$. The proposed scheme arrives at integrality authentication through the above way. At last, after checking that the $h(N_s+1)$ is equal to $C_4$, the *S* would replace the next time

authentication information $h^2(PW_i \oplus N_c^*)$ into the verification table. Therefore, it could ensure the authentication information $h^2(PW_i \oplus N_c^*)$ that is stored in the $SC_i$ as the same as stored in the verification table that would be stored in the $S$ to avoid DoS attack.

2. Prevent replay attack: because of the Nonce and the next time authentication information in the verification table that is stored by the $S$, the proposed scheme could conquer replay attack and the attacker is unable to pretend legal user for using the previous authentication information.
3. Prevent stolen-verifier attack: assume that the attacker steals the verification table from the $S$. Because of the $h()$ is a kind of one-way hash function, the attacker would not get the $h(PW_i \oplus N_c)$ and $PW_i$ from the authentication information $h^2(PW_i \oplus N_c)$. Similarly, the attacker would not know about the secret number $x$ that is stored by the $S$ to compute the correct authentication information $C_1$, $C_2$ and $C_3$. Therefore, the attacker is unable to pretend successful.
4. Prevent man-in-the-middle attack: if the attacker eavesdrops and records the authentication information $ID_i$, $C_1$, $C_2$, $C_3$ that the legal $U_i$ sends to the $S$ and the information $S_1$, $S_2$ that the $S$ returns to the $U_i$. Then, the attacker sends the $ID_i$, $C_{M1}$, $C_{M2}$, and $C_{M3}$ to the $S$, and $S_{M1}$, $S_{M2}$ to the $U_i$ respectively. Because the attacker does not know about the secret number $x$ that is stored by the $S$ and $PW_i$, $N_c$ that stored by the $SC_i$. The attacker would not to compute the $C_1$, $C_2$, $C_3$, $S_1$, and $S_2$ correctly. Then, the $S$ would reject provide the services and resources to the attacker and the $U_i$ would reject the services and resources that the attacker provides.
5. Prevent impersonation attack: because the attacker does not know about the secret number $x$ that stored in the $S$ and the verification information $h^2(PW_i \oplus N_c)$ that stored in the $SC_i$, therefore, the attacker is unable to pretend the legal user. If the attacker sends the $ID_i$ and pretension information $C_{M1}$, $C_{M2}$ and $C_{M3}$ to the $S$, the $S$ would compute the equations following:

   $C_{M1} \oplus h^2(x||ID_i) \oplus h^2(PW_i \oplus N_c) \rightarrow N_c^{**}$'

   $C_{M2} \oplus h(h^2(PW_i \oplus N_c) \oplus N_c^{**}) \rightarrow h^2(PW_i \oplus N_c^*)$'

   $C_{M3} \oplus h(h^2(PW_i \oplus N_c^*)) \rightarrow h(PW_i \oplus N_c)$'

   Then, the $S$ authenticate weather $h(h(PW_i \oplus N_c)')$ is equal to the value $h^2(PW_i \oplus N_c)$ that is stored in the verification table. But, the authentication information $C_{M1}$, $C_{M2}$ and $C_{M3}$ are not the correctly value. Therefore, the attacker is unable to pretend successful.
6. Prevent off-line password guessing attack: even though the attacker intercepts the login information $C_1$, $C_2$ and $C_3$ in the login phase, but the attacker does not know about the $N_c$, $N_c^*$ and secret number $x$. Therefore, the attacker is hard to use the dictionary to guess the $U_i$'s $PW_i$ correctly.

7. Prevent insider attack: in the register phase, the $U_i$'s system would use the $U_i$'s $PW_i$ and the random number $N_c$ to generate the $h^2(PW_i \oplus N_c)$ before send to the $S$. Because the insider is unable to obtain $PW_i$ with $h^2(PW_i \oplus N_c)$, if the user registers to other servers with the same $ID_i$ and $PW_i$, the insider is unable to use $h^2(PW_i \oplus N_c)$ to pretend this user successfully. Therefore, the insider could not login other servers to obtain the services or resources successfully.
8. Prevent smart card lose leading to the impersonation problem: if the $U_i$ loses the $SC_i$, the person who picks up the $SC_i$ and does not know about the legal $U_i$'s $PW_i$ is unable to compute the correct authentication information $C_1$, $C_2$ and $C_3$ to pretend the legal $U_i$ successfully.

## 5.2 The efficiency analysis

In this section, besides of analyzing the efficiency of our proposed scheme, we also compare with Chen's scheme and Wang's scheme in Table 1.

Table 1: The comparison table of efficiency analysis

| Scheme / Item | Chen's scheme (2003) | Wang's scheme (2004) | The proposed scheme |
|---|---|---|---|
| The Registration phase | $3T_h+3T_\oplus$ | $3T_h+1T_\oplus+2T_{\|\|}$ | $1T_e+1T_d+3T_h+2T_\oplus+1T_{\|\|}$ |
| The login and authentication and the key authentication phase | $9T_h+11T_\oplus$ | $12T_h+12T_\oplus+3T_{\|\|}$ | $16T_h+2T_F+18T_\oplus+1T_{\|\|}$ |
| The change password phase | × | × | $16T_h+2T_F+18T_\oplus+1T_{\|\|}$ |
| Total cost | $12T_h+14T_\oplus$ | $15T_h+13T_\oplus+5T_{\|\|}$ | $1T_e+1T_d+19T_h+2T_F$ $+20\ T_\oplus+2T_{\|\|}$ |

Notation description:
$T_e$：One Time Complexity of Public Key Encryption.
$T_d$：One Time Complexity of Public Key Decryption.
$T_h$：One Time Complexity of One-Way Hash Function.
$T_F$：One Time Complexity of F Function.
$T_\oplus$：One Time Complexity of XOR Operation.
$T_{||}$：One Time Complexity of String Concatenation Operation.

According to the above table, we can know the efficiency of the proposed scheme is the worse than the others. The main reason as follows:

1. To solve the unreasonable assumption of the secure channel: the proposed scheme adopts the Public Key Cryptosystem during the registration phase.
2. Provides the process to produce the session key: in order to produce the session key in the authentication and key agreement phase, it shall add 2 times *F()* complexity. Both user and server use *F()* to produce the same session key which is used for ensure the privacy and integrity of message after they authenticated each other.

In order to achieve the proposed of scheme's purposes, it really has paid more calculated cost. But, the paid is worth. In addition, the main calculation method of the proposed scheme is the hash function and XOR. Therefore, it still fit for lower computation and communication cost.

## 5.3 The security and integrity summary analysis

As for this section, we make the security and integrity analysis to show the schemes by Chen's, Wang's, and ours in Table 2.

Table 2: The comparison table of security and integrity analysis

| Schemes / Items | Chen (2003) | Wang (2004) | The proposed scheme |
|---|---|---|---|
| Reparability | △ | O | O |
| User freely choose password | O | O | O |
| Operation of the password change | △ | △ | O |
| Provide the mutual authentication | X | O | O |
| Provide the session key agreement | X | X | O |
| Assume the secure channel problem | X | X | O |
| Prevent the DoS attack | O | O | O |
| Prevent the replay attack | O | O | O |
| Prevent the stolen-verifier attack | O | O | O |
| Prevent the man-in-the-middle attack | O | O | O |
| Prevent the impersonation attack | O | O | O |
| Prevent the off-line password guessing attack | O | O | O |
| Prevent the insider attack | O | X | O |
| Prevent the smart card lose incur the impersonation problem | △ | O | O |
| △：unconsidered O：can prevent/provide X：can't prevent/provide | | | |

The above table shows the scheme comparison for security and integrity analysis. We can divide that into several parts and done elaboration.

1. Reparability: The scheme of the Wang's and the proposed scheme adopt the $SC_i$ to store the authentication information; only the Chen's scheme has not adopted the $SC_i$ as the storage. Hence, the Wang's and the proposed scheme need to consider this problem and achieves by the random number. Therefore, if the $U_i$ loses the $SC_i$ or when the $U_i$ thinks that the $SC_i$ is duplicated in the future, the $U_i$ could apply to reissue a new $SC_i$ to the $S$ again and needs not modify the identity of the $U_i$. Consequently, the Wang's and the proposed scheme satisfy the property of reparability.
2. Operation of the password change: the proposed scheme preserves the next authentication information with the verification table that is maintained by the $S$. These

would allow the $U_i$ to change his password through online. But, only the proposed scheme mentioned this topic.

3. Provide the mutual authentication: except for the $S$ authenticate the identity of the $U_i$, the proposed scheme should provide that the $U_i$ authenticate the identity of the $S$. Therefore, the scheme would avoid that the attacker pretend the identity of the $S$ for obtain the related information from the legal $U_i$, only the Chen's scheme without provide the mutual authentication.
4. Provide the session key agreement: after the $U_i$ has authenticated successfully in the authentication phase, the privacy and integrity of the transmitting information could protect with the production of the session key, and only the proposed scheme mentioned this topic.
5. Assume the secure channel problem: the schemes that the Chen and Wang proposed have this assumption during the register phase, but it is an unreasonable assumption actually. Therefore, the proposed scheme adopts the public key technology to solve this problem to avoid the unreasonable assumption.
6. Prevent insider attack: because the Wang's scheme has transmitted the $U_i$'s $ID_i$, $PW_i$ and the random number $N_c$ to the $S$ through a secure channel in the register phase. It would suffer from the insider of the $S$ records his $ID_i$ and $PW_i$. If the $U_i$ registers to the other servers with the same $ID_i$ and $PW_i$, then the insider could pretend this $U_i$ to login other servers successfully and obtain the services or resources.

# 6. Conclusion

A new password authentication scheme has been proposed in the paper that uses server's secret $x$ and the identity of user, and compared with the other previously proposed schemes. The proposed password authentication scheme achieves more functionality and security than the other scheme.

# References

1. Chen, Chien-Ming, "Attacks and Solutions on Hash-Based Password Authentication Protocols," Master Thesis, Department of Engineering Science, FU JEN Catholic University, (2003).
2. Chang, Chen-Cheng, Lin, Chu-hsing, and Chiang, Chi-han, 『Security of E-Commerce』, Taipei: UNALIS Corporation (2000).

3. Ku, Wei-Chi and Chen, Chien-Ming, "Stolen-Verifier Attack on Two New Strong-Password Authentication Protocols," *IEICE Transaction on Communications* (85:11) 2002, pp. 2519-2521.
4. Ku, Wei-Chi, Tsai, Hao-Chuan, and Chen, Shuai-Min "Two Simple Attacks on Lin-Shen-Hwang's Strong-Password Authentication Protocol," *ACM SIGOPS Operating System Review* (37:1) 2003, pp. 26-31.
5. Ku, Wei-Chi, Chen, Shuai-Min, and Tsai, Hao-Chuan, "Solutions on Hash-Based Password Authentication Protocols," *Communications of the CCISA* (9:3) 2003, pp. 32-42.
6. Lamport, L., "Password Authentication with Insecure Communication," *Communications of the ACM* (24) 1981, pp. 770-772.
7. Lin, Chin-Wei, Shen, Jau-Ji, and Hwang, Min-Shiang, "Security Enhancement for Optimal Strong-Password Authentication Protocol," *ACM SIGOPS Operating System Review* (37:3) 2003, pp.12-16.
8. Lin, Chun-Li, Sun, Hung-Min, and Hwang, Tzonelih, "Attacks and Solutions on Strong-Password Authentication," *IEICE Transactions on Communications* (84:9) 2001, pp. 2622-2627.
9. Sandirigama, Manjula, Shimizu, Akihiro, and Noda, Matu-Tarow, "Simple and Secure Password Authentication Protocol (SAS)," *IEICE Transactions on Communications* (83:6), 2000, pp. 1363-1365.
10. Shimizu, A., "A Dynamic Password Authentication Method by One-way Function," *IEICE Transactions on Information and Systems* (73:7) 1990, pp. 630-636.
11. Tsuji, Takasuke and Shimizu, Akihiro, "An Impersonation Attack on One-Time Password Authentication Protocol OSPA," *IEICE Transactions on Communications* (86:7) 2003, pp. 2182-2185.
12. Wang, Ren-Chiun, "A Study on Remote User Password Authentication Schemes," Master Thesis, Department of Information Management, Chaoyang University of Technology, 2004.
13. Network Security Technology Company Limited Website, http://www.nst.tw/html/products_cc1010.html, 2005.